



A Comparative Study of ID3 and MLP Algorithms

**VENKATA AKHIL
KARUMURI**

Department of CSE
GITAM University
Andhra Pradesh, India.

ROHITH SAJJA

Department of CSE
GITAM University
Andhra Pradesh, India.

SURESH BABU GONTLA

Department of IT
GITAM University
Andhra Pradesh, India.

PRUDHVI TEJA KONDAPARTHI

Department of IT
GITAM University
Andhra Pradesh, India.

VISHNU MURTHY

Department of CSE
GITAM University
Andhra Pradesh, India.

Abstract—Data mining on large databases has been a major concern in research community. To overcome the difficulty of analyzing huge amounts of data, classification is done on the datasets. In our project, we have used two classification algorithms ID3 and MLP. We focused on applying the concept of ID3 algorithm to find the nodes in the decision tree and back-propagation to calculate the weights of artificial neural network. These concepts are used for training the decision tree and artificial neural network classifiers for PIMA dataset. The classifiers thus obtained from training phase are tested. Then the corresponding accuracies are calculated and compared to find the efficient algorithm among ID3 and MLP.

Keywords- Decision Trees, ID3, Multi-Layer Perceptron, Artificial Neural Network, PIMA Dataset.

I. INTRODUCTION

With increase in the databases size in present world, analyzing the data in huge databases becomes difficult. Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Classification is a data mining function that assigns items in a collection to target categories or classes. Among many classification algorithms, we have taken ID3 and MLP algorithms for our work to compare.

In this paper, we have designed and trained decision tree and an artificial neural network which can classify unknown samples. We trained decision tree using ID3 algorithm and artificial neural network using back propagation (MLP) algorithm using sigmoid function as activation function. We tested the sample data using the previously trained decision tree, artificial neural networks. We compared the results obtained by using both the techniques. For testing and training the decision tree and artificial neural network, PIMA data set is taken.

In this project we train the decision tree using ID3 algorithm and artificial neural network using back-propagation algorithm. These classifiers are used for decision making for the datasets taken. The project starts with a literature study on training decision tree using ID3 algorithm and the artificial neural networks using back propagation. The training of the classifiers is done on PIMA dataset which is one amongst the datasets from UCI Machine Learning Repository. Later, testing is done on the neural network and decision tree, we conclude the project with comparison of testing results. The comparison

reveals that MLP is more efficient when compared to ID3 algorithm [1].

The rest of the paper is organized as follows: The detailed literature survey is discussed in Section 2. Section 3 discusses the methodology and process behind the work. Section 4 describes the results and observation. Section 5 concludes the work followed by future directions.

II. RELATED WORK

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks.

A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time. In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.

Classifications are discrete and do not imply order. Continuous, floating-point values would indicate a numerical, rather than a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm.

The simplest type of classification problem is binary classification. In binary classification, the

target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating.

In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown.

Classification models are tested by comparing the predicted values to known target values in a set of test data. The goal of the predictive models is to construct a model by using the results of the known data and is to predict the results of unknown data sets by using the constructed model. For instance a bank might have the necessary data about the loans given in the previous terms. In this data, independent variables are the characteristics of the loan granted clients and the dependent variable is whether the loan is paid back or not. The model constructed by this data is used in the prediction of whether the loan will be paid back by client in the next loan applications.

III. PROPOSED METHODOLOGY

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the file for “MSW_USltr_format”.

In this paper, an effort has been made to compare the performance of ID3 and multi-layer perceptron (MLP). Figure 1 explains the proposed methodology and flow of this section.

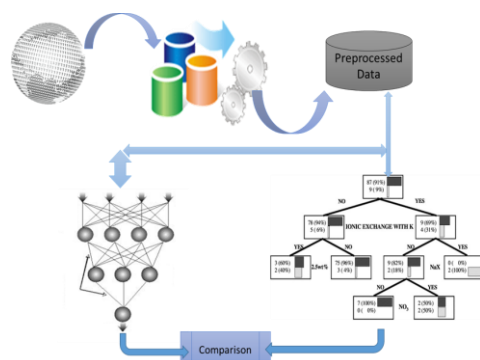


Figure 1: Process to group the dataset using k-means clustering.

The collected dataset is preprocessed and then it will be passed to two kinds of classifiers: ID3 and MLP. Later, the results are compared and observations are given.

A. Database

To train up the network we used PIMA dataset that contain 768 records and 8 attributes and one class variable. This data set collects information from patients who are all females over 21-year old of Pima Indian heritage. The attributes of the dataset are Number of times pregnant, Plasma glucose concentration a 2 hours in an oral glucose tolerance test, Diastolic blood pressure (mm Hg), Triceps skin fold thickness (mm), 2-Hour serum insulin (mu U/ml), Body mass index (weight in kg/(height in m)²), Diabetes pedigree function, Age (years) and Class variable (0 or 1). Of them 507(2/3rd of preprocessed dataset) are used for training and rest 254 (remaining 1/3rd of preprocessed dataset) are tested. The preprocessed dataset contain total of 7 attributes which includes 6 features and 1 class attribute. There 6 attributes are fed as inputs to the input layer [8].

B. Data Preprocessing

Before actual implementation, we need to preprocess the datasets in order to attain higher efficiencies of the classifiers. Artificial neural networks need tuples to be normalized before they are fed into the networks. After reading the data, tuples containing more number of null values are deleted. In the same way, attributed containing more than half null values are deleted. The maximum and minimum values of certain attributes are found out. Using these values, Min-max transformation is applied. Thus normalized datasets are used for neural network. However, decision tree does not require any normalization.

Preprocessing for decision tree is included in the actual implementation code.

Normalizing the required fields:

As it is difficult for computation with larger values during the training and testing of the neural network, generally the values are normalized between the range [0, 1]. So, if we want to normalize the field of the values then we have to find the maximum and minimum values of the field. The general formula for normalizing the value is

$$v' = \frac{(v - A_{min})}{(A_{max} - A_{min})}$$

Where, v' is normalized value, v is the old value, A_{min} is the minimum value of the attribute and A_{max} is the maximum value.

C. Decision Tree (ID3)

Decision trees are powerful and popular tools for classification and prediction. The attractiveness of

decision trees is due to the fact that, in contrast to neural networks, decision trees represent *rules*.

In some applications, the accuracy of a classification or prediction is the only thing that matters. In such situations we do not necessarily care how or why the model works. In other situations, the ability to explain the reason for a decision, is crucial. In marketing one has describe the customer segments to marketing professionals, so that they can utilize this knowledge in launching a successful marketing campaign. The domain experts must recognize and approve this discovered knowledge, and for this we need good descriptions. There are a variety of algorithms for building decision trees that share the desirable quality of interpretability [2].

Decision tree is a classifier in the form of a tree structure (see Figure 1), where each node is either:

- A leaf node - indicates the value of the target attribute (class) of examples, or
- A decision node - specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.

A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance.

Decision tree induction is a typical inductive approach to learn knowledge on classification. The key requirements to do mining with decision trees are:

- **Attribute-value description:** object or case must be expressible in terms of a fixed collection of properties or attributes. This means that we need to discretize continuous attributes, or this must have been provided in the algorithm.
- **Predefined classes (target attribute values):** The categories to which examples are to be assigned must have been established beforehand (supervised data).
- **Discrete classes:** A case does or does not belong to a particular class, and there must be more cases than classes.
- **Sufficient data:** Usually hundreds or even thousands of training cases.

Decision tree example

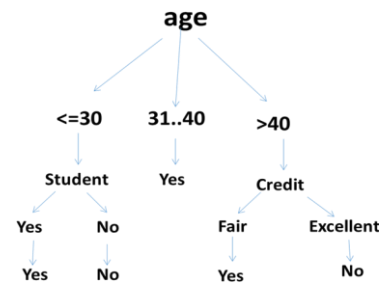


Figure 2: An example of a simple decision tree.

3.3.1. Construction of Decision Tree

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. Decision tree program construct a decision tree T from a set of training cases [3].

J. Ross Quinlan originally developed ID3 at the University of Sydney. He first presented ID3 in 1975 in a book, *Machine Learning*, vol. 1, no. 1. ID3 is based on the Concept Learning System (CLS) algorithm.

ID3 searches through the attributes of the training instances and extracts the attribute that best separates the given examples. If the attribute perfectly classifies the training sets then ID3 stops; otherwise it recursively operates on the m (where m = number of possible values of an attribute) partitioned subsets to get their "best" attribute. The algorithm uses a greedy search, that is, it picks the best attribute and never looks back to reconsider earlier choices. Note that ID3 may misclassify data [4].

The central focus of the decision tree growing algorithm is selecting which attribute to test at each node in the tree. For the selection of the attribute with the most inhomogeneous class distribution the algorithm uses the concept of entropy, which is explained next.

D. Artificial Neural Network

An Artificial Neural Network (ANN) is a highly parallel distributed network of connected processing units called neurons. It is motivated by the human cognitive process: the human brain is a highly complex, non-linear and parallel computer. The network has a series of external inputs and outputs which take or supply information to the surrounding environment. Inter-neuron connections are called synapses which have associated synaptic weights. These weights are also used to store knowledge which is acquired from the environment. Learning is achieved by adjusting

these weights in accordance with a learning algorithm. It is also possible for neurons to evolve by modifying their own topology which is motivated by the fact that neurons in the human brain can die and new synapses can grow. One of the primary aims of an ANN is to generalize its acquired knowledge to similar but unseen patterns. Two other advantages of biological neural systems are the relative speed with which they perform computations and their robustness in the face of environmental and/or internal degradation. Thus damage to a part of an ANN usually has little impact on its computational capacity as a whole. This also means that ANNs are able to cope with the corruption of incoming signals [5].

An Artificial Neural Network (ANN), usually called neural network (NN) is a mathematical model or a computational model that tries to simulate the structure and/or functional aspects of biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes the structure based on external or internal information that flows through the network during the learning phase. Neural networks are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. A neuron is an information processing unit that is fundamental to the operation of a neural network [6].

The word *network* in the term ‘artificial neural network’ arises because the function $f(x)$ is defined as composition of other functions $g(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as network’s structure with arrows depicting the dependencies between the variables. A widely used type of composition is a non-linear weighted sum,

$$f(x) = k * (\sum_i w_i g_i(x))$$

where

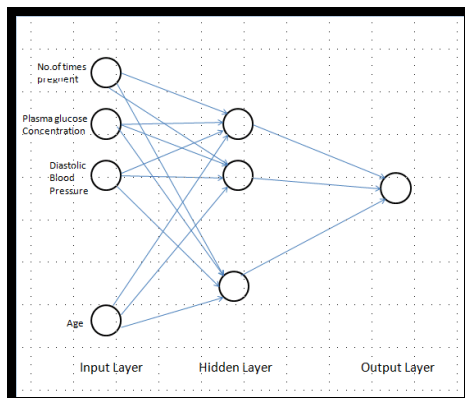


Figure 3: A simple representation of neural network.

Where k (commonly referred to as activation function) is some predefined function, such as hyperbolic tangent. It will be convenient for the following to refer to a collection of function g_i as simply a vector

$$g = (g_1, g_2, \dots, g_n)$$

Generally, in the traditional neural network the activation function of the units is sigmoid function

$$P(t) = \frac{1}{1 + e^{-t}}$$

where t , is the input to unit.

i. Construction of Decision Tree

The activation function, denoted by (v) , defines the output of neuron in terms of local field v . Three basic types of activation functions are as follows:

- **Threshold function (or Heaviside function):** A neuron employing this type of activation function is normally referred to as a McCulloch-Pitts model. The model has an all-or-none property
- **Piecewise –linear function:** This form of activation function may be viewed as an approximation to a non-linear amplifier. The following definition assumes the amplification factor inside the linear regions is unity.
- **Sigmoid function:** This is the most common form of activation function used in artificial neural network [7]. An example of a sigmoid function is the logistic function, defined by

$$P(t) = \frac{1}{1 + e^{-t}}$$

IV. RESULTS AND ANALYSIS

4.1. Confusion Matrix

A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a two class classifier.

Predicted class

Positive Negative

Actual class Positive A B

Negative C D

Table 1: A simple representation of confusion matrix.

The entries in the confusion matrix have the following meaning in the context of our study:

- a is the number of correct predictions that an instance is positive,
- b is the number of incorrect predictions that an instance is positive,
- c is the number of incorrect of predictions that an instance is negative, and
- d is the number of correct predictions that an instance is negative.

Several standard terms have been defined for the two class matrix:

- The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = (a+d)/(a+b+c+d)$$

- The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = a/(a+b)$$

- The false positive rate (FP) is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$FP = c/(c+d)$$

- The true negative rate (TN) is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$TN = d/(c+d)$$

- The false negative rate (FN) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = b/(a+b)$$

- Finally, precision (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = a/(a+c)$$

	Accuracy	True Positive Rate	False Positive Rate	Precision	Recall
Decision Tree(MATLAB)	68.5	56.62	25.56	48.56	56.62
MLP (MATLAB)	80.31	68.67	12.50	68.18	68.67
Decision Tree (WEKA)	78.34	55.12	14.60	64.38	55.12
MLP (WEKA)	75.55	96.77	24.15	57.00	96.77

Table 2. Comparative study of ID3 & MLP using WEKA and Matlab tool boxes for the PIMA dataset

Table 2. shows the comparative study for the results of ID3 and MLP. Matlab and WEKA tool boxes are used to do the analysis. True positive rate, false positive rate, precision, recall and overall accuracy are computed and shown in Table 2. From the analysis it is observed that, the accuracy is pretty good for MLP classifier which is developed in Matlab as compared to others.

V. CONCLUSION

In this paper, classifier with better accuracy, true positive rate are obtained when the classifier is trained with MLP instead of ID3 algorithm and when Matlab tool is used for the PIMA dataset. When the same dataset is given for training and testing in WEKA tool, we have observed that ID3 algorithm is better than MLP for PIMA dataset.

VI. REFERENCES

- [1]. Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications by S.Rajasekaran and G.A. Vijayalakshmi Pai.
- [2]. Jiawei Han and Micheline Kamber,"Data Mining Concepts and Techniques"-2ndEdition-Impressao
- [3]. Z. Haiyang, "A Short Introduction to Data Mining and Its Applications", IEEE, 2011
- [4]. J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986
- [5]. P Srinivasu, P S Avadhani, K Subhramanyam, B Sai Praveen, "Artificial neural Network Classifier for Intrusion Detection in Computer Networks"
- [6]. Ajit Abraham, Ravi. Jain, "Soft Computing Models For Network Intrusion Detection System".
- [7]. Zhi-Hua Zhou,Yuan Juang,"Mediacal Diagnosis with c4.5 Rule Preceded by Artificial Neural Network Ensemble",IEEE transactions on IT in Biomedicine.
- [8]. Cup, K. D. D. "Intrusion detection data set." (1999).